```
Set     Items   Description
S1       8149   (FIRST OR 1ST OR PRIME OR PRIMARY OR INITIAL OR ORIGINAL)(-
            2N)(INDEX? OR INDICES OR LIST? OR TREE?)
S2       6784   (SECOND? OR 2ND OR ADDITIONAL OR ANOTHER OR SUBSEQUENT)(2N-
            )(INDEX? OR INDICES OR LIST? OR TREE?)
S3      15591   (ONE OR SINGLE)(2N)(INDEX? OR INDICES OR LIST? OR TREE?)
S4    1370290   SEARCH? OR QUEST? OR PURSU? OR SEEK? OR QUER? OR MATCH?
S5     374368   REVIS? OR UPGRAD? OR UP()GRAD? OR UPDAT? OR UP()DAT?
S6    3580202   SWITCH? OR CHANG? OR SHIFT? OR INTERCHANG? OR TRADE? OR SU-
            BSTITUT?
S7    7702191   GENERATE? OR CREAT??? OR PRODUCE? OR DEVELOP? OR MAKE? ? OR
             ESTABLISH?
S8      50629   (STRUCTURALLY OR STRUCTURE? OR ARRANGEMENT? OR CONFIGURATI-
            ON? OR ORGANIZ?)(3N)(IDENTICAL OR MATCH? OR EXACT? OR SAME OR
            EQUAL OR CORRESPOND?)
S9    1497579   MAINTENANCE OR MAINTAIN? OR PRESERV? OR STABILITY OR PERMA-
            NENCE
S10      2748   TWO()(INDEX? OR INDICES OR LIST? OR TREE?)
S11        20   S9 (3N) S10
S12      1675   S1 AND S4
S13       303   S2 AND S5
S14         6   S12 AND S13
S15       226   S12 AND S6
S16         0   S15 AND S13
S17     15911   S7 AND S8
S18       937   S17 AND (INDEX? OR INDICES OR LIST? OR TREE?)
S19        20   S18 AND S1
S20         3   S19 AND S2
S21         8   S18 AND S2
S22    147337   S6 AND (INDEX? OR INDICES OR LIST? OR TREE?)
S23       189   S22 AND S17
S24        89   S5 (3N)(INDEX? OR INDICES OR LIST? OR TREE?) AND S1
S25       136   S11 OR S14 OR S19 OR S20 OR S21 OR S24
S26       125   S25 NOT PY>2001
S27       124   S26 NOT PD>20010103
S28       108   RD (unique items)
File   8:Ei Compendex(R) 1970-2004/Aug W4
         (c) 2004 Elsevier Eng.  Info. Inc.
File  35:Dissertation Abs Online 1861-2004/Jul
         (c) 2004 ProQuest Info&Learning
File 202:Info. Sci. & Tech. Abs. 1966-2004/Jul 12
         (c) 2004 EBSCO Publishing
File  65:Inside Conferences 1993-2004/Aug W5
         (c) 2004 BLDSC all rts. reserv.
File   2:INSPEC 1969-2004/Aug W4
         (c) 2004 Institution of Electrical Engineers
File 233:Internet & Personal Comp. Abs. 1981-2003/Sep
         (c) 2003 EBSCO Pub.
File  94:JICST-EPlus 1985-2004/Aug W1
         (c)2004 Japan Science and Tech Corp(JST)
File  99:Wilson Appl. Sci & Tech Abs 1983-2004/Jul
         (c) 2004 The HW Wilson Co.
File  95:TEME-Technology & Management 1989-2004/Jun W1
         (c) 2004 FIZ TECHNIK
File 583:Gale Group Globalbase(TM) 1986-2002/Dec 13
         (c) 2002 The Gale Group
```

28/5/3 (Item 3 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

05844705 E.I. No: EIP01266561364
Title: B+-tree indexes with hybrid row identifiers in Oracle8i**
Author: Chong, E.I.; Das, S.; Freiwald, C.; Srinivasan, J.; Yalamanchi, A.; Jagannath, M.; Tran, A.-T.; Krishnan, R.
Corporate Source: Oracle Corporation, Nashua, NH 03062, United States
Conference Title: 17th International Conference on Data Engineering
Conference Location: Heidelberg, Germany Conference Date: 20010402-20010406
Sponsor: IEEE; EML; IBM; SAS
E.I. Conference No.: 58203
Source: Proceedings - International Conference on Data Engineering 2001. p 341-348
Publication Year: 2001
CODEN: PIDEEG
Language: English
Document Type: CA; (Conference Article) Treatment: T; (Theoretical)
Journal Announcement: 0107W1
Abstract: Most commercial database systems support B**+-tree indexes using either 1) physical row identifiers, for example, DB2 or 2) logical row identifiers, for example, NonStop SQL. Physical row identifiers provide fast access to data. However, unlike logical row identifiers, they need to be updated whenever the row moves. This paper describes an alternate approach where hybrid row identifiers are used. A hybrid row identifier consists of two components: 1) a logical component, namely, the primary key of the base table row, and 2) a physical component, namely, the Database Block Address (DBA) of the row. By treating the DBA as a guess regarding where the row may be found, performance comparable to physical B**+-tree indexes is attained for valid guess-DBAs. This scheme retains the logical index advantage of avoiding an immediate **index update** when the base table row moves. Instead, an online utility can be used to lazily fix the invalid guess-DBAs. This scheme has been used to implement B**+-tree indexes for Oracle8i **index** -organized tables (**primary** B**+- **tree** like structure) which encounter both row movement and table reorganization. It can also be applied to conventional tables that undergo some form of row movement such as replication or transportation of data. This paper discusses the following: 1) key concepts for supporting B**+-tree indexes with hybrid row identifiers; 2) performance study comparing such indexes with physical B**+-tree indexes, and 3) applicability of B**+-tree indexes with hybrid row identifiers to speed up table replication (full-refresh), to increase index availability, and to support online table operations. 15 Refs.
Descriptors: *Database systems; Indexing (of information); Trees (mathematics)
Identifiers: Database block addresses (DBA)
Classification Codes:
723.3 (Database Systems); 903.1 (Information Sources & Analysis); 921.4 (Combinatorial Mathematics, Includes Graph Theory, Set Theory)
723 (Computer Software, Data Handling & Applications); 903 (Information Science); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 90 (ENGINEERING, GENERAL); 92 (ENGINEERING MATHEMATICS)

28/5/10 (Item 10 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

04493976 E.I. No: EIP96093330913
Title: Methodology for index configurations in object-oriented databases
Author: Seo, S.K.; Lee, Y.J.
Corporate Source: Korea Advanced Inst of Science and Technology, Taejon, South Korea
Source: Information Sciences v 93 n 3-4 Sep 1996. p 187-210
Publication Year: 1996
CODEN: ISIJBC ISSN: 0020-0255
Language: English
Document Type: JA; (Journal Article) Treatment: A; (Applications); T; (Theoretical)
Journal Announcement: 9611W1
Abstract: In this paper, we explore the problem of the optimal index configuration in object-oriented database systems. The problem is to find a set of nested attribute indexes such that they are used to minimize the overall cost of processing retrieval queries and maintaining the **indexes** for **update** queries in a given workload. We also take into account a space limit for storing the **indexes** . The **primary** idea of the proposed methodology is to reduce the problem to that of selecting indexes in such a way that the sum of cost savings is maximized subject to a given storage capacity. The correctness of our approach is proved using analytic cost formulas. We characterize the computational difficulty of the problem, which is shown to be NP-hard. We propose an effective approximation algorithm. The algorithm is compared with an optimal searching algorithm. Experiment results show the approximation algorithm performs well for various input situations. (Author abstract) 20 Refs.
Descriptors: *Database systems; Optimal systems; Maximum principle; Economics; Algorithms; Mathematical techniques
Identifiers: Index configuration; Nested attribute indexes
Classification Codes:
723.3 (Database Systems); 921.5 (Optimization Techniques); 921.2 (Calculus); 911.2 (Industrial Economics)
723 (Computer Software); 921 (Applied Mathematics); 911 (Industrial Economics)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS); 91 (ENGINEERING MANAGEMENT)

28/5/11 (Item 11 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

04230456 E.I. No: EIP95082827140
Title: Complete and recursive feature theory
Author: Backofen, Rolf; Smolka, Gert
Corporate Source: Universitaet des Saarlandes, Saarbruecken, Ger
Source: Theoretical Computer Science v 146 n 1-2 Jul 24 1995. p 243-268
Publication Year: 1995
CODEN: TCSCDI ISSN: 0304-3975
Language: English
Document Type: JA; (Journal Article) Treatment: T; (Theoretical)
Journal Announcement: 9510W3
Abstract: Various feature descriptions are being employed in logic programming languages and constraint-based grammar formalisms. The common notational primitive of these descriptions are functional attributes called features. The descriptions considered in this paper are the possibly quantified first-order formulae obtained from a signature of binary and

unary predicates called features and sorts, respectively. We **establish** a first-order theory FT by means of three axiom schemes, show its completeness, and construct three elementarily equivalent models. One of the models consists of the so-called feature graphs, a data structure common in computational linguistics. The other two models consist of the so-called feature **trees** , a record-like data **structure** generalizing the **trees** **corresponding** to **first** -order terms. Our completeness proof exhibits a terminating simplification system deciding validity and satisfiability of possibly quantified feature descriptions. (Author abstract) 23 Refs.

Descriptors: *Computation theory; Data structures; Computational grammars ; Computational linguistics; Logic programming; Constraint theory; Formal languages; Theorem proving; Computer programming languages

Identifiers: Feature descriptions; Feature graphs; Axiom schemes; Completeness proof

Classification Codes:

723.1.1 (Computer Programming Languages)

721.1 (Computer Theory, Includes Formal Logic, Automata Theory, Switching Theory, Programming Theory); 723.2 (Data Processing); 723.1 (Computer Programming)

721 (Computer Circuits & Logic Elements); 723 (Computer Software)

72 (COMPUTERS & DATA PROCESSING)

28/5/12 (Item 12 from file: 8)

DIALOG(R)File 8:Ei Compendex(R)

03782574 E.I. No: EIP94011179754

Title: Incremental distributed asynchronous algorithm for minimum spanning trees

Author: Tsin, Yung H.

Corporate Source: Univ of Windsor, Windsor, Ont, Can

Source: Computer Networks and ISDN Systems v 26 n 2 Oct 1993. p 227-232

Publication Year: 1993

CODEN: CNETDP ISSN: 0169-7552

Language: English

Document Type: JA; (Journal Article) Treatment: T; (Theoretical)

Journal Announcement: 9403W1

Abstract: A distributed algorithm for **updating** a minimum spanning **tree** when a new vertex is added to the underlying network (a connected, unidirected, weighted graph) is presented. The algorithm runs asynchronously, and the processor at each vertex of the network is required to know only information concerning its adjacent edges. The number of messages transmitted is bounded by 6N and the time complexity is O(H), where N and H(approximately equals N) are the number of vertices and the vertices and the height of the **original** minimum spanning **tree** , respectively. (Edited author abstract) 12 Refs.

Descriptors: *Computer networks; Algorithms; Trees (mathematics); Graph theory

Identifiers: Incremental distributed asynchronous algorithm; Minimum spanning trees; Distributed computer network

Classification Codes:

723 (Computer Software); 921 (Applied Mathematics)

72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

28/5/13 (Item 13 from file: 8)

DIALOG(R)File 8:Ei Compendex(R)

03536186 E.I. Monthly No: EI9301002203
Title: Automated verlet neighbor list algorithm with a multiple time-step approach for the simulation of large systems.
Author: Chialvo, Ariel A.; Debenedetti, Pablo G.
Corporate Source: Princeton Univ, Princeton, NJ, USA
Source: Computer Physics Communications v 70 n 3 Jul 1992 p 467-477
Publication Year: 1992
CODEN: CPHCBZ ISSN: 0010-4655
Language: English
Document Type: JA; (Journal Article) Treatment: T; (Theoretical); A; (Applications)
Journal Announcement: 9301
Abstract: The automated version of the Verlet neighbor list algorithm proposed by the authors left bracket Comput. Phys. Commun. 60 (1990) 215; 64 (1991) 15 right bracket is applied to the multiple time-step method left bracket Streett et al., Mol. Phys. 35 (1978) 639 right bracket . The algorithm, in which the optimum neighbor list thickness (SKIN) increases with the system size (N), and a **primary** neighbor **list** is **updated** at regular periods, regardless of N, shows a linear CPU time vs. N dependence over the range 10**3 less than N less than equivalent to 2 multiplied by 10**4. The algorithm's speed is 2.8 times higher than that achieved by the neighbor list algorithm with 'recommended' values for SKIN and update frequency left bracket Schoen, Comput. Phys. Commun. 52 (1989) 175 right bracket , using the same type of hardware. (Author abstract) 12 Refs.
Descriptors: *COMPUTER SIMULATION; ALGORITHMS; NUMERICAL ANALYSIS; OPTIMIZATION; LARGE SCALE SYSTEMS
Identifiers: VERLET NEIGHBOR LIST; MULTIPLE TIME-STEP METHOD
Classification Codes:
723 (Computer Software); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

28/5/14 (Item 14 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

03492080 E.I. Monthly No: EI9210123784
Title: Efficient distributed algorithm to solve updating **minimum spanning** tree **problem.**
Author: Park, Jungho; Hagihara, Ken'ichi; Tokura, Nobuki; Masuzawa, Toshimitsu
Corporate Source: Osaka, Univ, Toyonaka, Japan
Source: Systems and Computers in Japan v 23 n 3 1992 p 1-12
Publication Year: 1992
CODEN: SCJAEP ISSN: 0882-1666
Language: English
Document Type: JA; (Journal Article) Treatment: T; (Theoretical); A; (Applications)
Journal Announcement: 9210
Abstract: This paper proposes a distributed algorithm for reconstructing a minimum-weight spanning **tree** T **prime** of a network N prime when link addition and deletion occur in a network N with a minimum-weight spanning tree T. In this algorithm each processor uses information whose adjacent links belong to T in order to construct T prime efficiently. The communication complexity and ideal time complexity of the algorithm are O(n log(f plus t) plus m) and O(n log(f plus t) plus n, respectively, where n and e are the number of processors and that of links in N prime , t is the

Dynamic Response; MATHEMATICAL MODELS; COMPUTERS--Applications; AUTOMOBILES --Crashworthiness
Identifiers: COMBINED VEHICLE FRONT STRUCTURE; OCCUPANT DYNAMICS MODEL; LUMPED MASS STRUCTURE MODEL; CRASH ANALYSIS; STEERING COLUMN DISPLACEMENT
Classification Codes:
432 (Highway Transportation); 408 (Structural Design); 931 (Applied Physics); 921 (Applied Mathematics); 723 (Computer Software); 662 (Automotive Design & Manufacture)
43 (TRANSPORTATION); 40 (CIVIL ENGINEERING); 93 (ENGINEERING PHYSICS); 92 (ENGINEERING MATHEMATICS); 72 (COMPUTERS & DATA PROCESSING); 66 (AUTOMOTIVE ENGINEERING)

28/5/16 (Item 16 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

03324947 E.I. Monthly No: EI9111133795
Title: Binary search trees of almost optimal height.
Author: Andersson, Arne; Icking, Christian; Klein, Rolf; Ottmann, Thomas
Corporate Source: Lund Univ, Lund, Swed
Source: Acta Informatica v 28 n 2 Dec 1990 p 165-178
Publication Year: 1990
CODEN: AINFA2 ISSN: 0001-5903
Language: English
Document Type: JA; (Journal Article) Treatment: T; (Theoretical)
Journal Announcement: 9111
Abstract: First we present a generalization of symmetric binary B-trees, SBB(k)-trees. Whose properties and simplicity of implementation make it a useful alternative to other search trees in practical applications. Then, by using an SBB (k)-tree with a varying k we achieve a structure with a logarithmic amortized cost per update and a height of long n plus o(logn). This result is an improvement of the upper bound on the height of a dynamic binary search tree. By **maintaining two trees** simultaneously the amortized cost is transformed into a worst-case cost. Thus, we have improved the worst-case complexity of the dictionary problem. (Edited author abstract) 21 Refs.
Descriptors: *DATA PROCESSING--*Data Structures; MATHEMATICAL TECHNIQUES --Trees; COMPUTER METATHEORY--Computational Complexity; COMPUTER PROGRAMMING--Algorithms
Identifiers: BINARY SEARCH TREES; DICTIONARY PROBLEM; UPDATE ALGORITHMS
Classification Codes:
723 (Computer Software); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

28/5/17 (Item 17 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

03044129 E.I. Monthly No: EI9104038499
Title: Finding and updating **depth- first spanning trees of acyclic digraphs in parallel.**
Author: Chaudhuri, P.
Corporate Source: James Cook Univ of North Queensland, Townsville, Aust
Source: Computer Journal v 33 n 3 Jun 1990 p 247-251
Publication Year: 1990
CODEN: CMPJA6 ISSN: 0010-4620
Language: English

Document Type: JA; (Journal Article) Treatment: T; (Theoretical)
Journal Announcement: 9104
Abstract: Fast parallel algorithms are presented for finding and updating the depth- **first** spanning **tree** of an acyclic digraph. The machine model used is a parallel random access machine that allows simultaneous access to the same memory location during only the read operation. The parallel depth-first search algorithm is based on a partial spanning tree doubling technique which is found to be useful in updating the depth- **first** spanning **tree** when a new node (edge) is added to an acyclic digraph. The most notable result is an O(log n) time parallel algorithm for updating the depth- **first** spanning **tree** solution when a new node (edge) is added to an n node acyclic digraph whose depth- **first** spanning **tree** is known. (Author abstract) 14 Refs.
Descriptors: *COMPUTER SYSTEMS PROGRAMMING--*Multiprocessing Programs; MATHEMATICAL TECHNIQUES--Trees; COMPUTER PROGRAMMING--Algorithms
Identifiers: SPANNING TREES; ACYCLIC GRAPHS; DIRECTED GRAPHS; SEARCH METHODS; PARALLEL ALGORITHMS
Classification Codes:
723 (Computer Software); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

28/5/18 (Item 18 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

02965956 E.I. Monthly No: EI9010117531
Title: Influence of aggregate shape on base behavior.
Author: Barksdale, Richard D.; Itani, Samir Y.
Corporate Source: Georgia Inst of Technology, Atlanta, GA, USA
Source: Transportation Research Record n 1227 1989 p 173-182
Publication Year: 1989
CODEN: TRREDM ISSN: 0361-1981
Language: English
Document Type: RC; (Report Chapter) Treatment: X; (Experimental)
Journal Announcement: 9010
Abstract: A re-examination and simplification of the **original** rut **index** concept for predicting rut susceptibility in aggregate bases is presented to eliminate some of the disadvantages of the original approach. Variables investigated included density, gradation, moisture content, and aggregate shape and surface characteristics. The **revised** rut **index** concept was used to evaluate and compare the relative permanent deformation behavior of these various unbound aggregates. The cubic-shaped, smooth rounded river gravel was found to be more than two time as susceptible to rutting as the crushed aggregates tested. The crushed aggregates were angular, blade, and disc shaped and had relatively rough surfaces. These aggregates generally performed similarly with respect to permanent deformation, although the visual appearance of the two blade-shaped aggregates was not as nice as the others. The use of a simple, slow triaxial shear test as a practical alternative to the conventional dynamic test was studied for evaluating the resilient and permanent characteristics of unbound base materials. The slow triaxial shear test was found to be suitable for evaluating the resilient modulus, but appeared not to be appropriate for evaluating permanent deformation characteristics. (Edited author abstract) 27 Refs.
Descriptors: *GRANULAR MATERIALS--*Deformation; PAVEMENTS--Stresses; ROADBUILDING MATERIALS--Testing
Identifiers: RUT INDEX CONCEPT; AGGREGATE BASES; CYCLIC LOAD TESTS
Classification Codes:

483 (Soil Mechanics & Foundations); 406 (Highway Engineering); 421 (Materials Properties); 422 (Materials Testing)
48 (ENGINEERING GEOLOGY); 40 (CIVIL ENGINEERING); 42 (MATERIALS PROPERTIES & TESTING)

28/5/19 (Item 19 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

02587855 E.I. Monthly No: EI8806051942
Title: SENSITIVITY REDUCTION BY STATE DERIVATIVE FEEDBACK.
Author: Haraldsdottir, A.; Kabamba, P. T.; Ulsoy, A. G.
Corporate Source: Univ of Michigan, Ann Arbor, MI, USA
Source: Journal of Dynamic Systems, Measurement and Control, Transactions ASME v 110 n 1 Mar 1988 p 84-93
Publication Year: 1988
CODEN: JDSMAA ISSN: 0022-0434
Language: English
Document Type: JA; (Journal Article) Treatment: T; (Theoretical)
Journal Announcement: 8806
Abstract: This paper shows that the sensitivity of state feedback control systems can be reduced by additional state derivative feedback, for a fixed closed loop eigenstructure. The price of this sensitivity reduction is in general noise response amplification. **Two indices** which quantify **stability** robustness and response sensitivity are given for time invariant continuous time and discrete time systems, together with an index of response to disturbances and noise. Closed form expressions for the gradients of these indices are given. A two step design procedure is proposed which consists of first selecting a closed loop eigenstructure, then minimizing one of the sensitivity indices under a magnitude constraint on the noise response. Examples are given to illustrate this original design procedure. (Author abstract) 35 refs.
Descriptors: *CONTROL SYSTEMS--*Design; MATHEMATICAL TECHNIQUES--Eigenvalues and Eigenfunctions
Identifiers: SENSITIVITY REDUCTION; STATE DERIVATIVE FEEDBACK; FIXED CLOSED LOOP EIGENSTRUCTURE; FEEDBACK CONTROL SYSTEMS
Classification Codes:
731 (Automatic Control Principles); 601 (Mechanical Design); 921 (Applied Mathematics)
73 (CONTROL ENGINEERING); 60 (MECHANICAL ENGINEERING); 92 (ENGINEERING MATHEMATICS)

28/5/20 (Item 20 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

02187737 E.I. Monthly No: EI8704035331
Title: SEMI-DISTRIBUTED ADAPTIVE MODEL FOR REAL-TIME FLOOD FORECASTING.
Author: Corradini, Corrado; Melone, Florisa; Ubertini, Lucio
Corporate Source: IRPI, Perugia, Italy
Source: Water Resources Bulletin v 22 n 6 Dec 1986 p 1031-1038
Publication Year: 1986
CODEN: WARBAQ ISSN: 0043-1370
Language: ENGLISH
Document Type: JA; (Journal Article) Treatment: T; (Theoretical)
Journal Announcement: 8704
Abstract: A semi-distributed model for real-time flood forecasting in

28/5/24 (Item 24 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

00995941 E.I. Monthly No: EI8102012539 E.I. Yearly No: EI81023899
Title: 1-2 BROTHER TREES OR AVL TREES REVISITED.
Author: Ottmann, T.; Wood, D.
Corporate Source: Univ Karlsruhe, Ger
Source: Computer Journal v 23 n 3 Aug 1980 p 248-255
Publication Year: 1980
CODEN: CMPJA6 ISSN: 0010-4620
Language: ENGLISH
Journal Announcement: 8102
Abstract: 1-2 brother trees are binary **search** trees which have similarities to both height balanced **search** trees and 2-3 **trees**. **First**, 0(log//2n) insertion and deletion algorithms are demonstrated and their similarities with those for brother **trees** are noted. **Second**, it is proved that the space utilization of (random) 1-2 brother trees is much better than that for (random) 2-3 trees. Third, the close relationship between 1-2 brother trees and height balanced trees is demonstrated, and as this also holds for their right-sided counterparts is leads to 0(log//2n) insertion and deletion algorithms for right-sided height balanced trees. Finally, this demonstrates that the insertion and deletion algorithms for right-sided height balanced trees were already available, but hidden, in the corresponding algorithms for right brother trees. 14 refs.
Descriptors: *DATA PROCESSING--*Data Structures
Classification Codes:
723 (Computer Software)
72 (COMPUTERS & DATA PROCESSING)

28/5/25 (Item 25 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

00909240 E.I. Monthly No: EI8004030230 E.I. Yearly No: EI80044468
Title: INFORMATION RETRIEVAL WITH APL BY ADAPTIVE INDEX AND USER GUIDANCE.
Author: Schek, Hans-Joerg; Walch, Georg
Corporate Source: IBM, Heidelberg, Ger
Source: APL Quote Quad v 9 n 4 Pts 1 & 2 Jun 1979, APL '79 Conf Proc, Rochester, NY, May 30-Jun 1 1979 Publ by ACM, New York, NY, 1979 pt 1 p 385-392
Publication Year: 1979
Language: ENGLISH
Journal Announcement: 8004
Abstract: A system is described applicable for information retrieval and **update** both in formatted and unformatted files containing non-numerical data. It uses a new reference-string indexing technique which supports partial- **match** **queries** and similar-record **search**. The reference-string index is adapted to data usage or data. Those parts of records which are estimated by the program to be specified very often in **queries** are included as reference strings and inverted. For data access in the retrieval phase, the specified attribute values and their logical expressions are transformed into a logical expression between reference strings. **Primary** and **secondary** (**index**) data are stored as external

files using auxiliary processors for access. The retrieval and **update** functions are combined with the user-guidance component used to describe data (dictionary), to teach system use, and to assist as a permanently available help feature. As an internal structure it uses a semantic net to navigate the user from general to more specific information. A new interesting feature is the use of selected reference strings and selected combinations of them as a more detailed data description extracted automatically from the data or even data usage. 9 refs.

Descriptors: *INFORMATION RETRIEVAL SYSTEMS; COMPUTER PROGRAMMING LANGUAGES

Identifiers: APL LANGUAGE

Classification Codes:

901 (Engineering Profession); 723 (Computer Software)

90 (GENERAL ENGINEERING); 72 (COMPUTERS & DATA PROCESSING)

28/5/29 (Item 2 from file: 35)

01691473 ORDER NO: AAD99-21060

IMPROVING R-TREES FOR MULTIDIMENSIONAL QUERIES

Author: GARCIA ROJAS, YVAN JOSE

Degree: PH.D.

Year: 1999

Corporate Source/Institution: UNIVERSITY OF DENVER (0061)

Adviser: MARIO A. LOPEZ

Source: VOLUME 60/02-B OF DISSERTATION ABSTRACTS INTERNATIONAL. PAGE 712. 115 PAGES

Descriptors: COMPUTER SCIENCE

Descriptor Codes: 0984

Multidimensional queries such as geometrical intersection or containment arise in areas such as computer-aided design, geographic information systems, computer vision, temporal databases, and multi-keyed indexing for traditional databases. We consider large data sets that necessitate disk storage.

R-trees [Gut84] are efficient data structures for multidimensional queries on disk resident data. There are dynamic and static (packed) R-trees. Dynamic R-trees are more flexible since they accommodate **updates** , but packed R- **trees** are better in query performance and node utilization. In this dissertation, we present new algorithms to improve query performance (number of disk accesses) of both type of R- **trees** .

First , we show improvements on dynamic R-trees by an optimal bi-partition algorithm for R-trees insertion, and a new insertion framework. Finding optimal bi-partitions of rectangle sets affects query performance of dynamic R- **trees** . During **updates** , overflowed nodes need to be split while minimizing expected query time. We provide an algorithm for finding optimal bi-partitions for any function on bounding rectangles of resultant subsets. The algorithm runs in time <italic>O</italic>(<italic>n<super>d</super></italic>) (<italic>d </italic> = number of dimensions). Our experiments show the use of optimal splits results in 5% to 15% improvement. We complement our contribution with an R-tree insertion that uses global restructuring. Our insertion results in improvements by a factor between 1 and 2.2 over leading standards.

Next, we present a new R-tree packing algorithm. Our algorithm uses two guidelines: it partitions input data into subtrees in a top-down fashion, and it considers all cuts orthogonal to coordinate axes that result in packed trees to optimize an arbitrary objective function. Our

experiments indicate that for data with skew our algorithm results in trees that requires from 1 up to 3 times fewer disk accesses.

Finally, we develop a new node restructuring algorithm useful for post-optimization of existing R-trees or improvement of insertion algorithms. Our post-optimization improves query performance relative to the best packed R-trees by a factor between 1 and 2. We show how to improve insertion algorithms to produce R-trees whose query performance improves by a factor between 1 and 1.7 relative to best dynamic R-trees.

28/5/30 (Item 3 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online

01228045 ORDER NO: AADMM-61489
ON PARALLELIZING DEPTH FIRST **SEARCH** TREE **CONSTRUCTION OF AN UNDIRECTED GRAPH**
Author: ISAL, R. YUGO KARTONO
Degree: M.SC.
Year: 1990
Corporate Source/Institution: QUEEN'S UNIVERSITY AT KINGSTON (CANADA) (0283)
Source: VOLUME 30/03 of MASTERS ABSTRACTS.
PAGE 786. 98 PAGES
Descriptors: COMPUTER SCIENCE
Descriptor Codes: 0984
ISBN: 0-315-61489-7

Depth first search (DFS) is a versatile technique used in many sequential graph algorithms. The technique was first introduced by R. E. Tarjan in 1972, and has been used to solve many sequential graph problems. Unfortunately, finding a DFS tree of any given graph has been shown to be inherently sequential. Reif has proven that the ordered DFS problem is P-complete, for both directed and undirected graphs (Rei85). It is still an open problem whether the unordered DFS problem is in NC. For a general graph, there is no known NC algorithm for the unordered DFS problem. However, some NC algorithms for the unordered DFS problem are known when the input graphs are either directed acyclic graphs (Gho84, Kim86, Bon89) or planar graphs (Smi86, He87, Sha88). A random NC algorithm for finding an unordered DFS tree of an undirected graph can be found in (Agg88).

This thesis concentrates only on an unordered DFS tree of an undirected graph. This thesis has two main objectives. Firstly, to design an algorithm for **updating** a DFS **tree** of a given graph, when a new edge or a new vertex (together with some edges adjacent to it) is inserted into or deleted from the graph. Secondly, to attempt to find another method to construct a DFS tree of an undirected graph.

For the first objective, four NC algorithms to maintain a DFS tree on an edge or vertex addition or deletion are presented. All the algorithms run in O(log n) time by using O(n$\sp3$) processors on the CREW-PRAM model. For the second objective, a method to construct a DFS tree of an undirected graph is proposed. We conjecture that the algorithm runs in O(n log n) time by using O(n$\sp3$) processors on the CREW-PRAM model.

28/5/41 (Item 2 from file: 202)
DIALOG(R)File 202:Info. Sci. & Tech. Abs.

3102720

Methodology for index configurations in object-oriented databases.
Author(s): Seo, S K; Lee, Y J
Corporate Source: Korea Advanced Institute of Science and Tech., Taejon
Information Sciences vol. 93, no. 3/4, pages 187-210
Publication Date: Sep 1996
ISSN: 0020-0255
Language: English
Document Type: Journal Article
Record Type: Abstract
Journal Announcement: 3100

This paper explores the problem of the optimal index configuration in object-oriented database systems. The problem is to find a set of nested attribute indexes such that they are used to minimize the overall cost of processing retrieval queries and maintaining the **indexes** for **update** queries in a given workload. Also considered is a space limit for storing the **indexes** . The **primary** idea of the proposed methodology is to reduce the problem to that of selecting indexes in such a way that the sum of cost savings is maximized subject to a given storage capacity. The correctness of the approach is proved using analytic cost formulas. The authors characterize the computational difficulty of the problem, which is shown to be NP-hard. They propose an effective approximation algorithm. The algorithm is compared with an optimal searching algorithm. Experiment results show the approximation algorithm performs well for various input situations.

Descriptors: Algorithms; Databases; Indexes; Methodology
Classification Codes and Description: 6.02 (Bibliographic Search Services, Databases)
Main Heading: Information Systems and Applications

28/5/44 (Item 5 from file: 202)
DIALOG(R)File 202:Info. Sci. & Tech. Abs.

2602732
Binary search trees of almost optimal height.
Author(s): Andersson, A; Klein, R; Ottmann, T
Acta Informatica vol. 28, no. 2, pages 165-178
Publication Date: Dec 1990
ISSN: 0001-5903
Language: English
Document Type: Journal Article
Record Type: Abstract
Journal Announcement: 2600

A generalization of symmetric binary B-trees, SBB(k)-trees is presented. The maintenance algorithms require only a constant number of rotations per updating operation in the worst case. These properties together with the fact that the structure is relatively simple to implement makes it a useful alternative to other search trees in practical applications. Then, by using an SBB(k)-tree with a varying k a structure with a logarithmic amortized cost per update and a height of log n + o (log n) is derived. This result is an improvement of the upper bound on the height of a dynamic binary search tree. By **maintaining two trees** simultaneously the amortized cost is transformed into a worst-case cost. Thus, the worst-case complexity of the dictionary problem is improved.

Descriptors: Algorithms; Binary systems; Optimization; Searching
Classification Codes and Description: 5.06 (Software and Programming)
Main Heading: Information Processing and Control

28/5/45 (Item 6 from file: 202)
DIALOG(R)File 202:Info. Sci. & Tech. Abs.

2102602
Digital search trees revisited .
Author(s): Flajolet, P; Sedgewick, R
SIAM Journal on Computing vol. 15, no. 3, pages 748-767
Publication Date: Aug 1986
ISSN: 0097-5397
Language: English
Document Type: Journal Article
Record Type: Abstract
Journal Announcement: 2100

Several algorithms have been proposed which build search trees using digital properties of the search keys. A general approach to the study of the average case performance of such algorithms is discussed, with particular attention to the analysis of the digital search tree structures of Coffman and Eve. Specifically, the method leads to the solution of a problem left open by Knuth, finding the average number of nodes in digital search trees with both sons null. The paper may be of interest as a survey and tutorial treatment of the analysis of the three **primary** digital **tree** search methods: digital search trees, radix search trees, and Patricia trees.

Descriptors: Algorithms; Computing; Data structures; Performance
Classification Codes and Description: 5.11 (Searching and Retrieval)
Main Heading: Information Processing and Control

28/5/51 (Item 12 from file: 202)
DIALOG(R)File 202:Info. Sci. & Tech. Abs.

0801463
Machine-aided indexing.
Book Title: Report Ddc-tr-71-3. 1971 March. Defense Documentation Center, Alexandria, Virginia. 152 P. 8 Ref. Ntis: Ad-721 875; Hc $3.00, Mf $0.95. See Isa 72-2106/n.
Author(s): Klingbiel, Paul H
Publication Date: 1971
Language: English
Document Type: Book Chapter
Record Type: Abstract
Journal Announcement: 0800

Progress is reported on the development of a partial syntactic analysis technique for indexing text. Although over 500,000 words of text have been indexed, this report is limited to the analysis of results at the 115,000 word level. There is the expectation that the error rate of commission, the selection of grammatically incorrect word sequences, can be held to the 2 percent level. Dictionary growth is reasonable. Computer processing speeds are good. **Original** and **revised** **indexing** subroutines are provided.

Appendixes provide samples of good index terms for each acceptable format in the format dictionary as well as samples of incorrect word sequences.

Classification Codes and Description: 4.07 (Classification, Indexing, and Thesauri)
Main Heading: Information Recognition and Description

28/5/55 (Item 1 from file: 2)
DIALOG(R)File 2:INSPEC

6965803 INSPEC Abstract Number: C2001-08-4250-010
Title: Minimization of tree pattern queries
Author(s): Amer-Yahia, S.; Lakshmanan, L.V.S.; Sungran Cho; Srivastava, D.
Journal: SIGMOD Record Conference Title: SIGMOD Rec. (USA) vol.30, no.2 p.497-508
Publisher: ACM,
Publication Date: June 2001 Country of Publication: USA
CODEN: SRECD8 ISSN: 0163-5808
SICI: 0163-5808(200106)30:2L.497:MTPQ;1-1
Material Identity Number: A660-2001-003
Conference Title: 2001 ACM SIGMOD International Conference on Management of Data
Conference Date: 21-24 May 2001 Conference Location: Santa Barbara, CA, USA
Language: English Document Type: Conference Paper (PA); Journal Paper (JP)
Treatment: Theoretical (T)
Abstract: **Tree** patterns form a natural basis to query **tree** -structured data such as XML and LDAP. Since the efficiency of **tree** pattern **matching** against a **tree** - **structured** database depends on the size of the pattern, it is essential to identify and eliminate redundant nodes in the pattern and do so as quickly as possible. The authors study **tree** pattern minimization both in the absence and in the presence of integrity constraints (ICs) on the underlying **tree** -structured database. When no ICs are considered, we call the process of minimizing a **tree** pattern, constraint-independent minimization. We **develop** a polynomial time algorithm called CIM for this purpose. CIM's efficiency stems from two key properties: (i) a node cannot be redundant unless its children are, and (ii) the order of elimination of redundant nodes is immaterial. When ICs are considered for minimization, we refer to it as constraint-dependent minimization. For **tree** -structured databases, required child/descendant and type co-occurrence ICs are very natural. Under such ICs, we show that the minimal equivalent query is unique. We show the surprising result that the algorithm obtained by **first** augmenting the **tree** pattern using ICs, and then applying CIM, always finds the unique minimal equivalent query; we refer to this algorithm as ACIM. While ACIM is also polynomial time, it can be expensive in practice because of its inherent non-locality. We then present a fast algorithm, CDM, that identifies and eliminates local redundancies due to ICs, based on propagating "information labels" up the **tree** pattern. CDM can be applied prior to ACIM for improving the minimization efficiency. We complement our analytical results with an experimental study that shows the effectiveness of our **tree** pattern minimization techniques. (19 Refs)
Subfile: C
Descriptors: computational complexity; data integrity; minimisation;

pattern matching; query processing; **tree** data structures; **trees** (mathematics)

Identifiers: **tree** pattern query minimization; **tree** -structured data; XML; LDAP; **tree** pattern matching; **tree** -structured database; redundant nodes; integrity constraints; constraint-independent minimization; polynomial time algorithm; CIM; constraint-dependent minimization; **tree** -structured databases; child/descendant; type co-occurrence ICs; minimal equivalent query; unique minimal equivalent query; inherent non-locality; fast algorithm; information labels; minimization efficiency; analytical results; **tree** pattern minimization techniques

Class Codes: C4250 (Database theory); C6120 (File organisation); C6160 (Database management systems (DBMS)); C1160 (Combinatorial mathematics); C1180 (Optimisation techniques); C6130 (Data handling techniques); C4240C (Computational complexity)

28/5/56 (Item 2 from file: 2)

DIALOG(R)File 2:INSPEC

6912589 INSPEC Abstract Number: C2001-06-6160D-009

Title: B/sup +/-tree indexes with hybrid row identifiers in Oracle8i

Author(s): Eugene Inseok Chong; Souripriya Das; Yalamanchi, A.; Jagannath, M.; Freiwald, C.; Srinivasan, J.; Anh-Tuan Tran; Krishnan, R.

Author Affiliation: Oracle Corp., Nashua, NH, USA

Conference Title: Proceedings 17th International Conference on Data Engineering p.341-8

Publisher: IEEE Comput. Soc, Los Alamitos, CA, USA

Publication Date: 2001 Country of Publication: USA xxii+666 pp.

ISBN: 0 7695 1001 9 Material Identity Number: XX-2001-00441

U.S. Copyright Clearance Center Code: 1063-6382/2001/$10.00

Conference Title: Proceedings of 17th IEEE International Conference on Data Engineering

Conference Sponsor: IEEE Comput. Soc. Tech. Committee on Data Eng.; EML; IBM; Hewlett-Packard; SAS; Microsoft; ABB; Software AG; sd&m

Conference Date: 2-6 April 2001 Conference Location: Heidelberg, Germany

Language: English Document Type: Conference Paper (PA)

Treatment: Practical (P)

Abstract: Most commercial database systems support B/sup +/-tree indexes using either: physical row identifiers, for example, DB2; or logical row identifiers, for example, NonStop SQL. Physical row identifiers provide fast access to data. However, unlike logical row identifiers, they need to be updated whenever the row moves. This paper describes an alternate approach where hybrid row identifiers are used. A hybrid row identifier consists of two components: a logical component, namely, the primary key of the base table row; and a physical component, namely, the database block address (DBA) of the row. By treating the DBA as a guess regarding where the row may be found, performance comparable to physical B/sup +/-tree indexes is attained for valid guess-DBAs. This scheme retains the logical index advantage of avoiding an immediate **index update** when the base table row moves. Instead, an online utility can be used to lazily fix the invalid guess-DBAs. This scheme has been used to implement B/sup +/-tree indexes for Oracle8i **index** -organized tables (**primary** B/sup +/- **tree** like structure) which encounter both row movement and table reorganization. (15 Refs)

Subfile: C

Descriptors: database indexing; relational databases; software

performance evaluation; SQL; tree data structures
Identifiers: B+ tree indexes; hybrid row identifiers; Oracle8i; physical row identifiers; DB2; logical row identifier; NonStop SQL; database block address; index-organized tables; relational database
Class Codes: C6160D (Relational databases); C6120 (File organisation); C6140D (High level languages)

28/5/59 (Item 5 from file: 2)
DIALOG(R)File 2:INSPEC

6308627 INSPEC Abstract Number: C1999-09-6170K-026
Title: BOAT-optimistic decision tree construction
Author(s): Gehrke, J.; Ganti, V.; Ramakrishnan, R.; Wei-Yin Loh
Author Affiliation: Dept. of Comput. Sci., Wisconsin Univ., Madison, WI, USA
Journal: SIGMOD Record Conference Title: SIGMOD Rec. (USA) vol.28, no.2 p.169-80
Publisher: ACM,
Publication Date: June 1999 Country of Publication: USA
CODEN: SRECD8 ISSN: 0163-5808
SICI: 0163-5808(199906)28:2L.169:BODT;1-E
Material Identity Number: A660-1999-002
U.S. Copyright Clearance Center Code: 0163-5808/99/$05...$5.00
Conference Title: 1999 ACM SIGMOD International Conference on Management of Data
Conference Date: 1-3 June 1999 Conference Location: Philadelphia, PA, USA
Language: English Document Type: Conference Paper (PA); Journal Paper (JP)
Treatment: Practical (P)
Abstract: Classification is an important data mining problem. Given a training database of records, each tagged with a class label, the goal of classification is to build a concise model that can be used to predict the class label of future, unlabeled records. A very popular class of classifiers are decision trees. All current algorithms to construct decision trees, including all main-memory algorithms, make one scan over the training database per level of the tree. We introduce a new algorithm (BOAT) for decision tree construction that improves upon earlier algorithms in both performance and functionality. BOAT constructs several levels of the tree in only two scans over the training database, resulting in an average performance gain of 300% over previous work The key to this performance improvement is a novel optimistic approach to tree construction in which we construct an **initial tree** using a small subset of the data and refine it to arrive at the final tree. We guarantee that any difference with respect to the "real" tree (i.e., the tree that would be constructed by examining all the data in a traditional way) is detected and corrected. The correction step occasionally requires us to make additional scans over subsets of the data; typically, this situation rarely arises, and can be addressed with little added cost. Beyond offering faster tree construction, BOAT is the first scalable algorithm with the ability to incrementally **update** the **tree** with respect to both insertions and deletions over the dataset. This property is valuable in dynamic environments such as data warehouses, in which the training dataset changes over time. The BOAT update operation is much cheaper than completely rebuilding the tree, and the resulting tree is guaranteed to be identical to the tree that would be produced by a complete re-build. (25 Refs)

Subfile: C
Descriptors: data mining; decision trees; learning (artificial intelligence); optimisation; pattern classification
Identifiers: BOAT; optimistic decision tree construction; data mining problem; training database; class label; concise model; optimistic approach ; main-memory algorithms; scalable algorithm; incremental updating; dynamic environments; data warehouses; training dataset; update operation
Class Codes: C6170K (Knowledge engineering techniques); C6160 (Database management systems (DBMS)); C1160 (Combinatorial mathematics); C1250 (Pattern recognition); C1180 (Optimisation techniques)

28/5/60 (Item 6 from file: 2)
DIALOG(R)File 2:INSPEC

6219670 INSPEC Abstract Number: C1999-05-6160Z-007
Title: A generic approach to bulk loading multidimensional index structures
Author(s): van den Bercken, J.; Seeger, B.; Widmayer, P.
Author Affiliation: Fachgebiet Inf., Marburg Univ., Germany
Conference Title: Proceedings of the Twenty-Third International Conference on Very Large Databases p.406-15
Editor(s): Jarke, M.; Carey, M.; Dittrich, K.R.; Lockovsky, F.; Loucopoulos, P.; Jeusfeld, M.A.
Publisher: Morgan Kaufmann Publishers, San Francisco, CA, USA
Publication Date: 1997 Country of Publication: USA xvi+599 pp.
ISBN: 1 55860 470 7 Material Identity Number: XX-1997-02713
Conference Title: Proceedings of VLDB 97: 23rd International Conference on Very Large Databases
Conference Date: 26-29 Aug. 1997 Conference Location: Athens, Greece
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: Recently there has been an increasing interest in supporting bulk operations on multidimensional **index** structures. Bulk loading refers to the process of **creating** an **initial index** structure for a presumably very large data set. We present a generic algorithm for bulk loading which is applicable to a broad class of **index** structures. Our approach differs completely from previous ones for the following reasons. First, sorting multidimensional data according to a predefined global ordering is completely avoided. Instead, our approach is based on the standard routines for splitting and merging pages which are already fully implemented in the **corresponding index structure** . **Second** , in contrast to inserting records one by one, our approach is based on the idea of inserting multiple records simultaneously. As an example we demonstrate how to apply our technique to the R- **tree** family. For R- **trees** we show that the I/O performance of our generic algorithm meets the lower bound of external sorting. Empirical results demonstrate that performance improvements are also achieved in practice without sacrificing query performance. (20 Refs)
Subfile: C
Descriptors: database **indexing** ; query processing; sorting; **tree** data structures; **trees** (mathematics); very large databases
Identifiers: generic approach; bulk loading multidimensional **index** structures; bulk operations; **initial index** structure; very large data set; generic algorithm; multidimensional data sorting; predefined global ordering; standard routines; multiple records; R- **tree** family; I/O

performance; external sorting; performance improvements; query performance
  Class Codes: C6160Z (Other DBMS); C1160 (Combinatorial mathematics); C6120 (File organisation); C6130 (Data handling techniques); C4250 (Database theory)
  Copyright 1999, IEE

28/5/64 (Item 10 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 2004 Institution of Electrical Engineers. All rts. reserv.

5377897 INSPEC Abstract Number: C9611-6160J-002
Title: Methodology for index configurations in object-oriented databases
  Author(s): Sang Koo Seo; Yoon Joon Lee
  Author Affiliation: Dept. of Comput. Sci., Korea Adv. Inst. of Sci. & Technol., Seoul, South Korea
  Journal: Information Sciences vol.93, no.3-4 p.187-210
  Publisher: Elsevier,
  Publication Date: Sept. 1996 Country of Publication: USA
  CODEN: ISIJBC ISSN: 0020-0255
  SICI: 0020-0255(199609)93:3/4L.187:MICO;1-6
  Material Identity Number: I132-96011
  U.S. Copyright Clearance Center Code: 0020-0255/96/$15.00
  Document Number: S0020-0255(96)00057-6
  Language: English Document Type: Journal Paper (JP)
  Treatment: Practical (P)
  Abstract: We explore the problem of the optimal index configuration in object-oriented database systems. The problem is to find a set of nested attribute indexes such that they are used to minimize the overall cost of processing retrieval queries and maintaining the **indexes** for **update** queries in a given workload. We also take into account a space limit for storing the **indexes** . The **primary** idea of the proposed methodology is to reduce the problem to that of selecting indexes in such a way that the sum of cost savings is maximized subject to a given storage capacity. The correctness of our approach is proved using analytic cost formulas. We characterize the computational difficulty of the problem, which is shown to be NP-hard. We propose an effective approximation algorithm. The algorithm is compared with an optimal searching algorithm. Experiment results show that the approximation algorithm performs well for various input situations. (20 Refs)
  Subfile: C
  Descriptors: computational complexity; indexing; object-oriented databases; search problems
  Identifiers: index configurations methodology; object-oriented databases; nested attribute indexes; retrieval queries; update queries; storage capacity; NP-hard; optimal searching algorithm
  Class Codes: C6160J (Object-oriented databases); C4240C (Computational complexity); C1180 (Optimisation techniques); C1160 (Combinatorial mathematics)
  Copyright 1996, IEE

28/5/67 (Item 13 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 2004 Institution of Electrical Engineers. All rts. reserv.

4705854 INSPEC Abstract Number: C9408-7250-011
Title: Random sampling from pseudo-ranked B/sup +/ trees
  Author(s): Antoshenkov, G.

Author Affiliation: Data Base Syst. Group, Digital Equipment Corp., Nashua, NH, USA
p.375-82
Editor(s): Li-Yan Yuan
Publisher: Morgan Kaufmannn, San Mateo, CA, USA
Publication Date: 1992 Country of Publication: USA xiii+631 pp.
Conference Title: Proceedings of 18th International Conference on Very Large Data Bases
Conference Sponsor: IEEE; ACM; VLDB Endowment; Canadian Informat. Processing Soc.; et al
Conference Date: 23-27 Aug. 1992 Conference Location: Vancouver, BC, Canada
Language: English Document Type: Conference Paper (PA)
Treatment: Theoretical (T)
Abstract: In the past, two basic approaches for sampling from B+ trees have been suggested: sampling from the ranked trees and acceptance/rejection sampling from non-ranked **trees** . The **first** approach requires the entire root-to-leaf path to be updated with each insertion and deletion. The second has no update overhead, but incurs a high rejection rate for the compressed-key B+ trees commonly used in practice. We introduce a new sampling method based on pseudo-ranked B+ trees, which are B+ trees supplemented with information loosely describing the estimated rank limits. This new method exhibits a very small rejection rate while paying only a marginal cost of the **tree** **update** overhead. We also present comparative efficiency measurements of different methods that were run on production databases and on several prototype workload simulations. (15 Refs)
Subfile: C
Descriptors: information retrieval; statistical analysis; trees (mathematics); very large databases
Identifiers: random sampling; pseudo-ranked B/sup +/ trees; ranked trees; acceptance/rejection sampling; non-ranked trees; root-to-leaf path; compressed-key B+ trees; efficiency measurements; roduction databases; prototype workload simulations
Class Codes: C7250 (Information storage and retrieval); C6160Z (Other DBMS); C1160 (Combinatorial mathematics); C1140Z (Other and miscellaneous)

28/5/80 (Item 26 from file: 2)
DIALOG(R)File 2:INSPEC

00038240 INSPEC Abstract Number: C69005735
Title: Variable length tree structures having minimum average search time
Author(s): Patt, Y.N.
Author Affiliation: US Army Research Office, Durham, NC, USA
Journal: Communications of the ACM vol.12, no.2 p.72-6
Publication Date: Feb. 1969 Country of Publication: USA
CODEN: CACMA2 ISSN: 0001-0782
Language: English Document Type: Journal Paper (JP)
Abstract: Considers the organisation of a file as a doubly-chained tree structure if it is necessary both to search and to **update** frequently. **First** , **trees** which have the property that a priori the filial set of each node is well defined are studied. It is proved that coding the nodes within each filial set with respect to the number of terminal nodes reachable from each is necessary and sufficient to guarantee minimum average search time. Then the more general case (that is, where the entire structure of the tree is changeable) is treated. A procedure is developed for constructing a tree with a minimum average search time. A simple closed

expression for this minimum average search time is obtained as a function of the number of terminal nodes. The storage capacity required to implement the doubly- chained tree structure on a digital computer is also determined.
Subfile: C
Descriptors: data structures; file organisation; information retrieval; trees (mathematics)
Class Codes: C6120 (File organisation)

28/5/82 (Item 2 from file: 233)
DIALOG(R)File 233:Internet & Personal Comp. Abs.

00616486 00DD12-005
Virtual lists **for Win32 -- Speeding** up database **searches**
Sipe, Steve
Dr. Dobb's Journal , December 1, 2000 , v23 n12 p56-61, 3 Page(s)
ISSN: 1044-789X
Languages: English
Document Type: Articles, News & Columns
Geographic Location: United States
Describes CVListCtrl, a class that works in conjunction with the listview control (LC) to provide a virtual list (VL) implementation. Explains that a VL contains only a few visible items at any one time and relies on callback routines to retrieve additional items from a secondary storage as needed. Indicates that to create a virtual **listview** , one **first** needs to add the LC to the dialog, then change the listview's style to show that it is a virtual listview. Adds that once a VL control has been created, one must set the total count of entries that the VL contains, so the VL can properly set the range of scrollbars and properly retrieve each item of information through callbacks. Covers the need for the loading process, using index numbers to request list entries, positioning the database's cursor to the record that corresponds to a specified list index, and advanced features such as caching and searching. (jb)
Descriptors: Information Retrieval; Database; Search Engines; Database Management; Index; Caching; Information Science

28/5/83 (Item 3 from file: 233)
DIALOG(R)File 233:Internet & Personal Comp. Abs.

00399178 95PW10-017
ZyIndex
Miastkowski, Stan
PC World , October 1, 1995 , v13 n10 p76, 1 Page(s)
ISSN: 0737-8939
Company Name: ZyLab
Product Name: ZyIndex for Windows
Languages: English
Document Type: Software Review
Grade (of Product Reviewed): B
Hardware/Software Compatibility: IBM PC Compatible; Microsoft Windows
Geographic Location: United States
Presents a favorable review of ZyIndex 5.2 for Windows ($395), a text indexing program from ZyLab (800). The program consists of two modules, ZyBuild and ZyFind. ZyBuild indexes all words in specified files or in all files. When it **first** creates an **index** it requires quite a bit of time

(30 minutes for 14MB of data) and **indexes** must be **updated** regularly, but ZyBuild can do updates in the background. Since all words are indexed, ZyFind can locate desired words very quickly. The user enters a word or Boolean expression and in only seconds the program provides a list of ``hits'' which are files that can be viewed, printed, or launched in their related apps. The program also supports ``fuzzy'' searches. Calls the interface in the ZyBuild module ``frequently esoteric'' and says it needs work but the program is capable of handling large amounts of information. (djd)

Descriptors: Indexing; Software Review; Window Software; Utility Program

Identifiers: ZyIndex for Windows; ZyLab

28/5/85 (Item 5 from file: 233)

DIALOG(R)File 233:Internet & Personal Comp. Abs.

00160126 88PI01-013

Memory Lane keeps an index for quick file searches

Mendelson, Edward

PC Magazine, Jan 12 1988, v7 n1 p48, 1 Pages

ISSN: 0745-2500

Languages: English

Document Type: Software Review

Grade (of Product Reviewed): B

Hardware/Software Compatibility: IBM PC; IBM PC Compatible

Geographic Location: United States

Presents a favorable review of Memory Lane v. 1.2 ($99), a memory resident indexing and search retrieval utility, from Group L Corp. of Herndon, VA (703). Requires 100K and DOS 2.0, and runs on an IBM PC or compatible. States that it automatically **updates** the **index** . It will retrieve any text from inside an application. Says that once the **initial indexing** is completed, the actual searches take around five seconds. (tjm)

Descriptors: UTILITY PROGRAM; SOFTWARE REVIEW; **INDEXING** ; INFORMATION RETRIEVAL; **UPGRADE**

Identifiers: Memory Lane; Group L

28/5/93 (Item 7 from file: 94)

DIALOG(R)File 94:JICST-EPlus

01367802 JICST ACCESSION NUMBER: 91A0469882 FILE SEGMENT: JICST-E

Weighted shortest path tree problem and breadth first search tree updating problem in networks.

MASUZAWA TOSHIMITSU (1); MIURA KOJI (1); TOKURA NOBUKI (1); PARK J (2)

(1) Osaka Univ., Faculty of Engineering Science; (2) Sunghwa Univ.

Denshi Joho Tsushin Gakkai Gijutsu Kenkyu Hokoku(IEIC Technical Report (Institute of Electronics, Information and Communication Enginners), 1991, VOL.91,NO.20(COMP91 1-11), PAGE.51-60, FIG.4, TBL.2, REF.8

JOURNAL NUMBER: S0532BBG

UNIVERSAL DECIMAL CLASSIFICATION: 65.012.122:519.86/.87

LANGUAGE: Japanese COUNTRY OF PUBLICATION: Japan

DOCUMENT TYPE: Journal

ARTICLE TYPE: Original paper

MEDIA TYPE: Printed Publication

ABSTRACT: A BFST updating problem is a problem to reconstruct a new BFST